iniciación al basic



ANAYA
MULTIMEDIA



El cubo informático

Iniciación al basic

Depósito Legal: M. 35.804 - 1985
ISBN: 84-7614-054-1
Printed in Spain
Imprime: Josmar, S. A. - Artesanía, 17
Polígono Industrial de Coslada (Madrid)

Iniciación al basic



ÍNDICE



EL CUBO INFORMÁTICO
DE ERNESTO DELTECLADO

¡Hola! Ese muñeco tan divertido que se pasea por los micro-ordenadores con unas gafas grandes como pantallas, es, Ernesto Delteclado. Le conocimos por casualidad una tarde, llevaba un mando de juego en una mano y una "cassette" en la otra. En un principio parecía un poco empollón, pero resultó ser muy simpático en cuanto abordamos el tema de la informática:

"Contrariamente a lo que se piensa, un micro-ordenador no es una caja mágica que sepa de todo, comprenda cualquier cosa y resuelva todos los problemas. Sigue siendo una máquina inerte que se anima cuando se pulsa un interruptor, pero incapaz del menor esfuerzo si no interviene el hombre... afortunadamente."

Después de esas generalidades, nos propuso iniciarnos al lenguaje BASIC con explicaciones, ejercicios, simulación y juegos, y sacó un cubo de su bolsillo. Ese cubo lo tenéis ahora en la mano, acompañado de un librito de fichas de trabajo sacado de su experiencia...

Lo más extraño es que no volvimos a ver a Ernesto desde aquella tarde. Cuentan que vive en los ordenadores... Ridículo. Sin embargo, algunas veces le volvemos a ver al frotar el cubo. ¡Y si la fábula de Aladino funcionase en el siglo XX!

ORGANIZACIÓN DEL LIBRITO

Este librito se compone de fichas todas iguales, compuestas de la siguiente manera:

- TÍTULO .... N.º ☐
  - Objeto.
  - Palabras clave.
  - Algunos detalles.
  - Ejercicios y juegos.

- **TITULO**: Los principales capítulos de la iniciación (ver Índice) para los distintos temas.
- **N.º** ☐ : Está en blanco a propósito para escribir en él los números del contador de la "cassette" que correspondan a los problemas.
- **OBJETO**: Expone la naturaleza del problema presentado.
- **PALABRAS CLAVE**: Permite la localización rápida, en caso de desear buscar un concepto concreto. Estas palabras clave vienen repetidas en el Anexo.
- **ALGUNOS DETALLES**: Son apuntes complementarios a las explicaciones de los programas: puntualizaciones, astucias.

- **EJERCICIOS Y JUEGOS**: Dan informaciones sobre las reglas y el tipo de ejercicios propuestos para asimilar mejor las nociones estudiadas.

Esas fichas están estructuradas por temas y aunque están unidas entre sí, se pueden utilizar separadamente (para repasar un tema hay que coger la "cassette" a partir de la cifra que hayas apuntado en la ficha).

Al final del librito encontrarás en el Anexo las palabras clave, un índice y unos consejos que proponen un balance de la iniciación (programación y corrección programa).

Los programas y el libro se completan; algunas explicaciones están en la pantalla, otras en las fichas y lo mismo sucede con los ejercicios.

El mejor método para conseguir una buena asimilación de esta iniciación es tener el librito al alcance de la mano mientras se siga el desarrollo de los programas en la pantalla.

Al final de cada demostración aparece en la pantalla la pregunta: ¿Deseas repasar estas nociones? S/N. Si tecleas S, el ordenador repite las explicaciones; si tecleas N aparece el mensaje: "Deja avanzar la cassette" y se carga el

programa siguiente. Te aconsejamos que apuntes el número que indica en aquel momento el contador de tu "cassette" de forma que sólo tengas que colocar correctamente la cinta para cualquier estudio ulterior.

CÓMO UTILIZAR EL CUBO

Este cubo informático se compone de 4 "cassettes". Cada "cassette" tiene varios programas unidos entre sí. Cada "cassette" se reconoce por su número de orden lógico.

MODO DE EMPLEO

Este programa funciona con la configuración siguiente:

— Unidad central MSX con una capacidad de memoria viva al menos de 16 K, bien con teclado AZERTY, bien con teclado QWERTY.

— TV doméstico o monitor con su cable.

— Lector grabador de programas.

— Cable de conexión.

Asegúrate que todos estos elementos están correctamente conectados y enchufados a la red.

1. Conecta el lector grabador de programas a la unidad central MSX.

2. Enchufa a la red la unidad central.

3. Introduce la "cassette" en el lector, comprobando que la "cassette" está totalmente rebobinada.

Si tu "cassette" tiene control remoto, pulsa la tecla PLAY del "cassette" y el ordenador se encargará de poner en marcha o detener la "cassette", en el momento deseado, evitando así cualquier otra manipulación.

Si tu "cassette" no tiene control remoto, tendrás que pulsar la tecla PLAY del mismo, cada vez que aparece en la pantalla el mensaje "DEJA AVANZAR LA CASSETTE".

En cuanto termina la carga del programa y empieza su ejecución, pulsa la tecla STOP del "cassette", si es que no tienes control remoto.

4. Teclea RUN "CAS" y pulsa la tecla RETURN (según los modelos RETURN o ⏎).

Puesto que todos los programas están unidos entre sí, el programa asegura la carga y el lanzamiento de los programas uno tras otro.

Sólo basta que te preocupes del "cassette", si no dispones de control remoto.

CÓMO FUNCIONA

N.º ☐

LAS DISTINTAS PARTES DE TU MICRO-ORDENADOR

SIMULACIONES

- DE LA CARGA DE UN PROGRAMA
- DE LA EJECUCION DE UN PROGRAMA

PALABRAS CLAVE

UNIDAD CENTRAL, TECLADO, PANTALLA, "CASSETTE".

ALGUNOS DETALLES

Este primer programa presenta las distintas partes de tu micro-ordenador, Mirémoslo detalladamente:

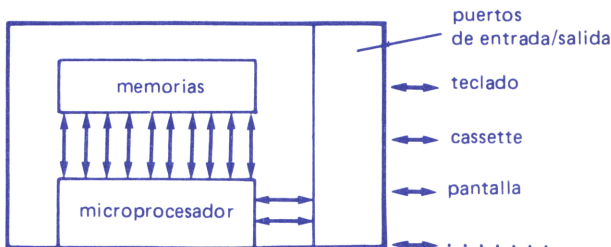
UNIDAD CENTRAL: Es la parte más vital de un ordenador. Contiene los componentes electrónicos que constituyen el cerebro de la máquina:

— El MICROPROCESADOR que gestiona y selecciona las informaciones.

— Las MEMORIAS que las almacenan.

— Los PUERTOS DE ENTRADA SALIDA que permiten la comunicación con el MICRO-PROCESADOR, por el teclado, la "cassette" o la pantalla.

Lo que se resume en el siguiente esquema:



PANTALLA: Es tu televisor o un monitor. Tú le transmites señales a través de la entrada PERITEL o UHF (toma de antena). Estas señales controlan los haces luminosos que crean la imagen en la pantalla.

"CASSETTE" (o lector de disquetes): Sirve para almacenar programas de forma permanente en un soporte magnético (por ejemplo, el CUBO INFORMÁTICO). Por lo cual, después de crear un programa, es preferible SALVAGUARDARLO en una "cassette" y evitar así tener que volverlo a escribir cuando tu ordenador está ya apagado. Otro método de almacenamiento más industrial es el CARTUCHO que tiene componentes en los cuales están grabados los datos.

Estamos hablando de informaciones.
¿Qué son realmente?

Un ordenador sólo se compone de señales elementales: sí o no que son BITS ("binar y digit"). Sólo puede distinguir estas dos opciones. Por eso, todas las órdenes que le transmitirás siguen un código llamado binario (0 y 1, sí y no). El código de los números es el siguiente:

Número	Código
0	0
1	1
2	10
3	11
4	100
5	101

Una serie de 0 y 1 equivale a un número, el ordenador lee un tren de ocho 0 ó 1: es un OCTETO.

En realidad, en memoria y en las cintas magnéticas, las informaciones son series de 0 y 1.

EJERCICIOS Y JUEGOS

Intentemos imaginar la traducción de informaciones según un código binario, seleccionando el número de BIT necesario:

Ejemplo: Elegir entre tres opciones, queso, postre, café. No se puede elegir sólo entre 0 ó 1, hacen falta dos niveles de elección (2 bits).

Queso:	*0 0*
Postre:	*0 1*
Café:	*1 0*

Otro ejemplo: En una clase, seleccionemos los chicos morenos de ojos verdes.

Podemos utilizar 3 bits:

moreno < *sí* / *no*

chico < *sí* / *no*

ojos verdes < *sí* / *no*

De esta forma, todos los alumnos podrán tener un código del tipo:

0 1 0 =chico que no tiene los ojos verdes y no es moreno.
1 0 0 =chica morena que no tiene los ojos verdes...

Podemos clasificar los alumnos con 3 bits.

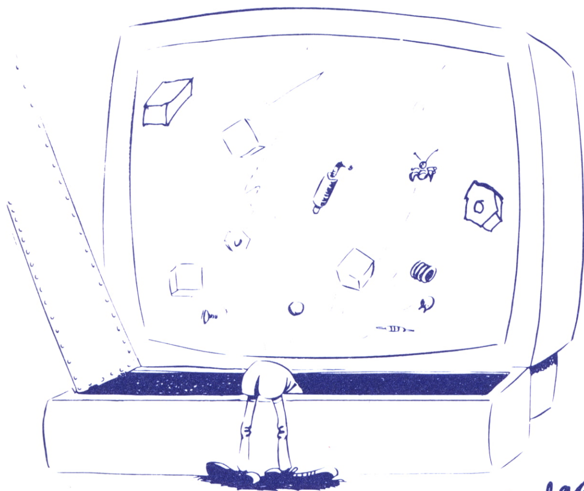
Siguiendo este mismo principio, a ver si consigues imaginar las codificaciones para los problemas siguientes:

— Se mezclaron accidentalmente una bolsa de judías pintas y otra de judías blancas y hay que separarlas.

— ¿Sabrás distinguir los reinos vegetal, animal y mineral?

— Hay que comprobar que todas las luces de una casa de cinco habitaciones están apagadas.

Todos los elementos de tu micro-ordenador están unidos entre sí y las transmisiones de informaciones se hacen por medio de cables dentro de los cuales circulan esos "trenes" eléctricos. Sigue la pequeña animación para entender el orden de estas transmisiones.



el teclado

EL TECLADO
2 programas

N.º ☐

PRESENTACIÓN DEL TECLADO.

RECONOCIMIENTO DE LAS TECLAS.

FUNCIONAMIENTO DEL TECLADO.

ACOPLAMIENTO DE LAS TECLAS.

APRENDIZAJE DEL TECLEO EN EL TECLADO.

PALABRAS CLAVE

TECLADO, TECLAS, MINÚSCULAS, EDITOR, CONTROL, COMUNICACIÓN.

ALGUNOS DETALLES

PROGRAMA 1

¿Para qué sirve un ordenador doméstico? : permite un diálogo entre el hombre y la máquina. Mientras esperamos la transmisión de órdenes orales o la conexión directa cerebro humano-microprocesador... tenemos el teclado para dictar nuestras observaciones a los circuitos electrónicos.

Descubramos entonces la geografía del teclado, a través de un juego de reconocimiento de posición de las teclas.

En un primer tiempo, los caracteres no aparecen en la pantalla y los estudiaremos a continuación. Tienes que reaccionar rápidamente puesto que si la contestación no está dada en pocos segundos, el ordenador la ofrece.

PROGRAMA 2

Aquí tienes el teclado presentado con todas sus teclas. Se distinguen varios tipos de teclas:

— Las TECLAS DE CONTROL que envían directamente una orden al micro-ordenador:

STOP, detiene el desarrollo de un programa.

CTRL combinado con una tecla alfabética tiene una acción específica:

CTRL + STOP interrumpe la ejecución de un programa.

CTRL + E borra la línea que estaba escribiéndose a partir de la posición del cursor.

— Las TECLAS DE FUNCIONES son una de las peculiaridades de los ordenadores MSX. El usuario dispone de 10 teclas programables a las cuales puede asignar las funciones que desea. Esas teclas tienen unas funciones determinadas por falta de asignación en el momento de enchufar la máquina a la red. Se sitúan en la parte superior del teclado y se identifican con los símbolos F1 - F2 - F3 - F4 - F5. Se obtienen las cinco siguientes pulsando simultáneamente la tecla SHIFT y una de las teclas citadas antes.

Por ejemplo, si pulsamos SHIFT y F1 simultáneamente, el usuario accede a la función F6.

De la misma manera, SHIFT y F5 dan acceso a la función F 10.

Para más comodidad, las funciones de cada tecla programable están siempre escritas en la parte inferior de la pantalla.

– Las TECLAS DE CARACTERES sirven para escribir las letras, las cifras o caracteres especiales en la pantalla. La combinación SHIFT y A producirá la aparición de una A mayúscula si el teclado se encuentra en modo minúscula o la aparición de una a minúscula si el teclado se encuentra en modo mayúscula.

– Las TECLAS DE EDICIÓN permiten el desplazamiento del cursor y las modificaciones de las impresiones en la pantalla o de las líneas BASIC.

Se pueden insertar caracteres por medio de la tecla INS o borrar otros por medio de la tecla DEL.

Y, por fin, una tecla importante: RETURN (o ⏎ en ciertos MSX); una simple presión sobre esta tecla produce la transmisión al microprocesador de la frase tecleada. En un primer momento, cuando tecleas algo en tu teclado, el ordenador lo ignora si no viene convalidado por RETURN.

De la misma manera, si modificas una línea BASIC aparecida en pantalla sin convalidar de nuevo la línea, no se tendrá en cuenta esta modificación.

EJERCICIOS Y JUEGOS

El ordenador te propone un entrenamiento al reconocimiento del teclado que consiste en copiar algunas frases. Cuidado, que todos los fallos son detectados incluidos los de modo (minúscula/mayúscula).

¡Al teclado! ... ¡Te vas a sorprender!

DATOS Y VARIABLES
3 programas

N.º ☐

NOCIÓN DE DATOS.

CÓDIGOS DE CARACTERES SEGUN LAS NORMAS ASCII.

VARIABLES NUMÉRICAS Y ALFANUMÉRICAS.

TRATAMIENTO DE DATOS EN MEMORIA.

PALABRAS CLAVE

DATOS, VARIABLE, CARÁCTER, ASCII, SIGNO =

ALGUNOS DETALLES

PROGRAMA 1

La definición de lo que es un dato sigue siendo la tarea más difícil del profesor. En efecto, el ordenador no sabe gestionar más que trenes de 8 informaciones elementales del tipo 10101001. Matemáticamente se demuestra fácilmente que existen 256 posibilidades de combinar 8 informaciones elementales 0 ó 1. El ordenador sabe distinguir 256 informaciones distintas de base. Cada constructor tiene la posibilidad de decidir que 00000001 representará para su ordenador el carácter "A", otro constructor elegirá la codificación 00001000 para el carácter "A". Con el fin de evitar este tipo de problema existe una norma internacional llamada norma ASCII que determina la correspondencia entre las distintas combinaciones de 0 y de 1, y los caracteres alfabéticos. El programa te permite conocer las codificaciones elegidas por la norma ASCII y cómo el ordenador almacena las informaciones en memoria, en función de la norma. Esta norma consta de tres partes.

Cada octeto de la memoria de tu ordenador se compone de 8 bits (informaciones elementales 0 ó 1), lo que teóricamente permite distinguir 256 informaciones distintas. Para su buen funcionamiento, el ordenador necesita poder distinguir 32 códigos de mando que no nombraremos aquí; representan los 32 primeros códigos ASCII de 0 a 31. Los códigos de 32 a 128 como los podrás constatar en el programa, representan todos los caracteres alfabéticos en mayúsculas y minúsculas, más algunos caracteres especiales.

Todos estos detalles están destinados para ayudarte a entender la noción de datos tal y como la concibe el ordenador: sólo entiende los octetos. Cuando introducimos un dato en la memoria del ordenador, éste lo clasifica según su costumbre. Por ejemplo, un dato alfanumérico "ABA" pasará primero a los códigos ASCII de las letras que lo componen es decir: "65,66,65", y después, cada código ASCII se colocará tal cual en la memoria.

Para que el ordenador pueda distinguir en el tratamiento una variable alfanumérica, se les añade un sufijo: el signo $. Por ejemplo:

MA$ representa una variable alfanumérica, TOTO representa una variable numérica.

que hace con ellas tu ordenador: las guarda en memoria y puede utilizarlas, sumarlas y transformarlas.

Una simulación te mete de lleno dentro de esta memoria. El microprocesador coge unos valores en memoria, opera con ellos y los vuelve a guardar en memoria según las instrucciones programadas.

Este intermedio técnico es necesario para una buena comprensión de los signos algebraicos tradicionales:

= es una equivalencia: en una casilla llamada A está el valor 32. A = 32. A$ = "CALE" significa que en las casillas A$, encontramos los valores 67/65/76/69.

+ **es una reunión:** en la casilla C están los valores de A y B: C = A + B.

En la casilla C$ se pone el contenido de la casilla A$ (HOLA) y de la casilla B$ (SEÑOR):

C$ = A$ + B$; en C$ encontraremos HOLA SEÑOR.

Esta última operación se llama CONCATENACIÓN y le dedicaremos un capítulo ulterior.

Una variable para el ordenador es una zona de memoria que en el instante T posee un contenido numérico o alfanum´ alfanumérico. Algunas operaciones son posibles en esas zonas de memoria y es lo que indica la potencia de un ordenador.

Evidentemente, estas operaciones son muy distintas según el tipo de variables y resulta inconcebible dividir una variable alfanumérica por un número. Si tenemos A$ = "CUBO", eso significa que el ordenador ordena en una de sus zonas de memoria, los valores de los caracteres ASCII que componen la palabra CUBO. Pero para el ordenador, resultará más fácil manipular A$ que "CUBO".

En resumen, un dato es la materia prima que un programa va a manipular siguiendo unas órdenes precisas. Expondremos esto más detalladamente en el próximo capítulo.

PROGRAMA 2

Después de haber aprendido a reconocer y a distinguir los distintos tipos de variables, intentemos entender ahora lo

EJERCICIOS Y JUEGOS

PROGRAMA 3

El programa siguiente se dedica a hacerte jugar a reconocer las variables y los datos. Consiste en hacer funcionar una fábrica llevando las mercancías adecuadas a los talleres correspondientes. Para lo cual te ayudarás de las teclas flechadas para hacer bajar, subir, ir a izquierda o a derecha en su carrito. La tecla F te permite interrumpir el juego.

ANÁLISIS N.º ☐

NOCION DE PROGRAMA.

PRESENTACION DEL BASIC.

ANALISIS DE UN PROGRAMA

PALABRAS CLAVE

BASIC, PROGRAMA, ANALISIS, COMUNICACION.

ALGUNOS DETALLES

Ahora estamos listos para entrar en el meollo del tema. Un ordenador sólo ejecuta las acciones para las cuales se le programa. Si deseas cambiar el color de la pantalla, debes permitir al ordenador unas cuantas órdenes que cumplirá.

Hemos visto que las informaciones que entiende la máquina son muy elementales (trenes de ocho 0 ó 1). No se trata de escribir ese tipo de sucesión de cifras, el teclado no serviría de nada, bastaría un contacto (como para el morse). Se utilizan más frecuentemente unos lenguajes de programación y el BASIC (Beginners All Purpose Simbolic Instruction Code, código de instrucciones simbólicas para principiantes) es uno de los más accesibles. Se compone de palabras sacadas del inglés que permiten definir instrucciones (PRINT = visualización en pantalla, LIST = listado del programa en pantalla, RUN = ejecución de un programa...)

¡Así la conversación se hace más rápida!

Esas órdenes o instrucciones definen unas cuantas tareas elementales que constituyen los elementos del programa. Hemos simbolizado esos elementos por un tubular compuesto de pequeñas estructuras o tubos que se pueden organizar como se quiera. ¿Por qué unos tubulares? Pues porque de la misma manera que el agua y la arena circulan dentro, los datos son orientados y modificados según los programas.

De hecho, el trabajo del ordenador consistirá en reconocer,

manipular, seleccionar, expulsar unos datos (cifras o letras) para llevarlos de un estado A (salida) a un estado B (llegada). Por ejemplo, si un programa calcula un tasa del 15 por 100, coge el valor y restituye el resultado. También se podría multiplicar el valor inicial por 3, añadir 3, dividir por 3, restar 1, volver a tomar el valor, multiplicar por 0,85, restar el valor inicial..., para obtener el mismo resultado. La descomposición de un problema en tareas elementales pertenece al programador; es él quién tiene que analizar una acción para descomponerla lo más racionalmente posible.

En resumen, podemos decir que cualquier problema se puede dividir en subtareas elementales para que el ordenador las trate: esto es el **ANÁLISIS de un problema.**

EJERCICIOS Y JUEGOS

Te proponemos un breve ejercicio de análisis de tarea:

Debes dar la vuelta de una compra de 80F, el cliente te da 100F. En caja tienes 3 monedas de 5F, 5 monedas de 2F, 10 monedas de 1F. El cliente te pide que le devuelvas la menor cantidad de monedas posible.

Tendrás que buscar una vía de análisis del problema y proponer además una solución divertida.

ORDINOGRAMAS N.º ☐

NOCIONES DE ORDINOGRAMAS.

PALABRAS CLAVE

ORDINOGRAMAS, ANÁLISIS.

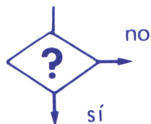
ALGUNOS DETALLES

El ordenador nos va a obligar a analizar lo que queremos ver ejecutar. Para lo cual existe un código de análisis que permite realizar esquemas: los ordinogramas.

Las acciones vienen señaladas por rectángulos:



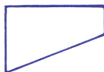
Las preguntas por rombos (siempre sí o no)



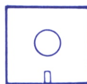
Las flechas simbolizan las funciones entre tareas.
Las salidas son representadas por pictogramas.



PANTALLA



IMPRESORA



DISCO



"CASSETTE"

Esta racionalización permite sistematizar el análisis con el fin de asociar una tarea elemental con una instrucción.

En la pantalla, puedes ver la asociación de ordinogramas y de instrucciones.

Esas instrucciones se presentan de la manera siguiente (ejemplo):

120 PRINT "HOLA"

120 representa la dirección de la instrucción y PRINT "HOLA" es el texto.

Las instrucciones están en memoria según su dirección, salvo si el texto de la instrucción lo decide de otra forma.

La instrucción 2 se ejecuta después de la instrucción 1, la 3.ª después de la 2.ª, etc.

Tenemos que precisar sucesivamente, todos los textos posibles para las instrucciones.

Esos textos forman parte de las funciones BASIC.

Realmente todo el secreto del BASIC reside en los pocos párrafos que vienen a continuación. El resto es presentación, astucia y elegancia.

Existen 6 tipos de órdenes fundamentales en BASIC:

La orden de consulta **PRINT** (visualización)

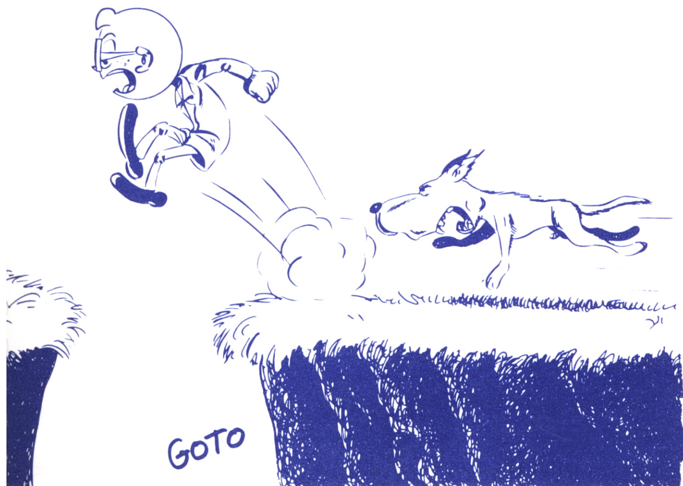
Permite la visualización en la pantalla de una cadena de caracteres de cualquier tipo o de números. Su sintaxis es: PRINT "HOLA" o PRINT 123 o PRINT A

No olvides las comillas para las palabras o las cadenas de caracteres. La ejecución de esta instrucción provoca la visualización en la pantalla de la frase puesta entre comillas o del número del valor de la variable.

Si tienes prisa, puedes teclear una señal de interrogación en vez de PRINT porque el BASIC lo acepta como una a abreviatura de la palabra PRINT.

La orden de adquisición **INPUT** (entrada)

Esa orden autoriza la aceptación de un valor o de una palabra escrita por el usuario en el teclado. Para calcular una tasa sobre un valor a introducir utilizaremos este tipo de orden que significa un poco: ¿qué valor o qué palabra?


GOTO

Su sintaxis es:

INPUT A para una aceptación de número.

INPUT A$ para una aceptación de cadena de caracteres "cassette" número 2).

Cuando tu microordenador encuentra tal instrucción, se pone en espera de una intervención exterior, lo que se traduce por un signo de interrogación seguido de un cursor parpadeante.

La orden de salto **GOTO**

Sabemos que un programa se desarrolla según el orden creciente de las direcciones de las instrucciones. El usuario puede, sin embargo, modificar tal ejecución pidiendo al ordenador que se dirija a otra dirección.

Para eso hay que utilizar GOTO que tiene la síntaxis siguiente:

GOTO xxx donde xxx representa la dirección en la cual queremos que el ordenador siga ejecutando el programa.

La dirección especificada puede ser anterior o posterior a la dirección donde se encuentra la instrucción GOTO.

La orden de test IF... THEN...

El ordenador sólo sabe reconocer valores binarios. Lo mismo sucede con sus decisiones, no conoce el "tal vez": una condición está realizada o no. Es cierto que parece un poco maniqueo pero nadie es perfecto. Su síntaxis es la siguiente:

IF (condición) THEN xxxx

La condición es una afirmación que requiere una respuesta entre dos posibles o bien la condición se comprueba, o bien lo contrario.

Por ejemplo IF A = 2 o IF A$ = "HOLA".

xxxx representa la dirección a la cual se debe dirigir el ordenador si la condición se verifica. En caso contrario, el ordenador ejecuta la instrucción siguiente IF... THEN...

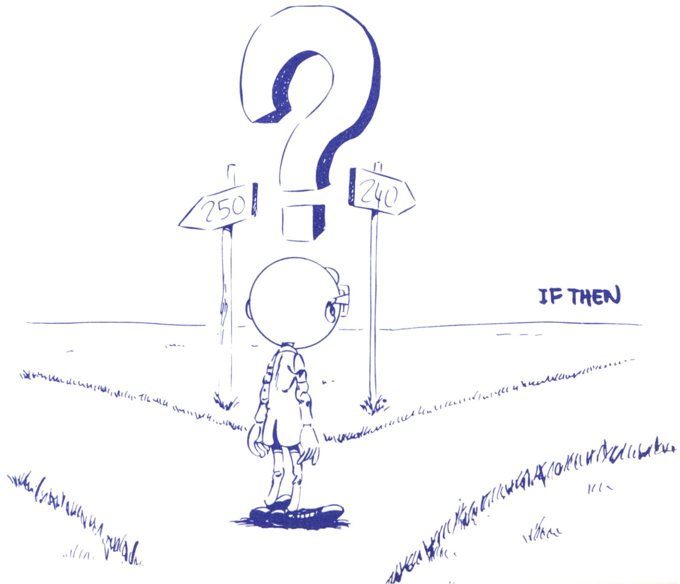
Se puede también utilizar la instrucción bajo la forma:
IF (condición) THEN (acción).

No basta entonces con indicar al ordenador adonde se tiene que dirigir: le pediremos que actúe inmediatamente.

Por ejemplo:

IF X=4 THEN PRINT "HOLA"

En un primer tiempo te aconsejamos que utilices esta instrucción en su primera forma, con el fin de facilitar la puesta a punto y la claridad de tu programa.



La orden de repetición FOR... NEXT...

Quizá sea la instrucción original de la informática.

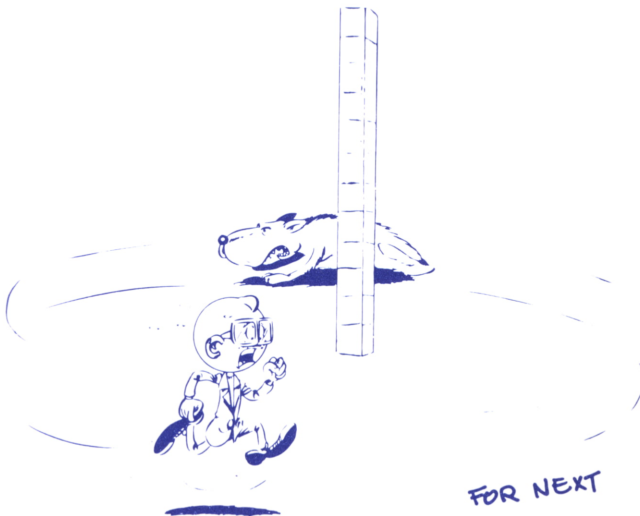
Es la orden que permite hace repetir una acción unas cuantas veces antes de pasar a la ejecución del resto del programa.

Su sintaxis es:

```
FOR I = 1 TO 5
Acción
NEXT 1
```

Se descompone en 2 líneas que indican el número de repeticiones a ejecutar. En nuestro ejemplo 5 veces, por una acción dada. I es el indicador del bucle, representa el indicador que mide el número de veces que se repitió la instrucción.

Cada vez que el ordenador encuentra la segunda parte de la instrucción NEXT I, aumenta el valor de I en 1 y ejecuta otra vez la acción requerida a condición de que el segundo valor límite estipulado en FOR I = 1 TO N no sea alcanzado si no se saldría del bucle.


FOR NEXT

Las órdenes **DATA** y **READ**

Se utilizan conjuntamente para almacenar datos en un programa. Esos datos pueden ser números o elementos de texto. Las sintaxis son:

DATA 1789, REVOLUCION FRANCESA
1515 MARIGNANO
1944 DESEMBARCO
READ A,B$,C,D$,E,F$

Después de la ejecución de la instrucción de la orden DATA, el ordenador guarda en memoria las informaciones que queríamos memorizar y sabe donde están. La instrucción READ (leer) permite ir a buscar esas informaciones.

Si se da la instrucción: PRINT A,B$,C,D$,E,F$ obtendremos:

1789 REVOLUCION FRANCESA
1515 MARIGNANO
1944 DESEMBARCO

Las comas situadas entre las variables, las separan y permiten al ordenador reconocerlas.

Una instrucción DATA se puede insertar en cualquier momento del programa y el ordenador sabrá encontrarla en su momento.

Existen todavía muchas instrucciones del BASIC más o menos importantes cuyas utilizaciones vendrán explicadas a lo largo de todo el cubo. Por ejemplo CLS que borra la pantalla o PLAY "CD" que tocará las dos notas de música DO y RE.

Te aconsejamos que consultes el manual de utilización de tu micro-ordenador que te facilitará una lista completa de las instrucciones utilizables.



EJERCICIOS Y JUEGOS

Te proponemos varios ejercicios para concluir este capítulo. Intenta analizar algunos problemas y redactar los ordinogramas correspondientes:

– Tú eres guardabarrera. Cómo organizar el movimiento de tu barrera.

– Estás jugando con un dado. En tres tiradas tienes que totalizar al menos 12 puntos, pero no más. Puedes parar cuando lo desees.

– Estás vendiendo tomates a 6,78 Pta el kilo. ¿Qué programa te permitirá tener el precio de venta en función del peso de mercancía vendida?

EJECUCIÓN DE UN PROGRAMA

N.º □

1 programa

DESARROLLO DE UN PROGRAMA.

SIMULACIÓN DE SU EJECUCIÓN INTERNA Y EXTERNA.

ASOCIACIÓN INSTRUCCIÓN-EFECTO.

PALABRAS CLAVE

LISTADO, ANÁLISIS, SIMULACIÓN.

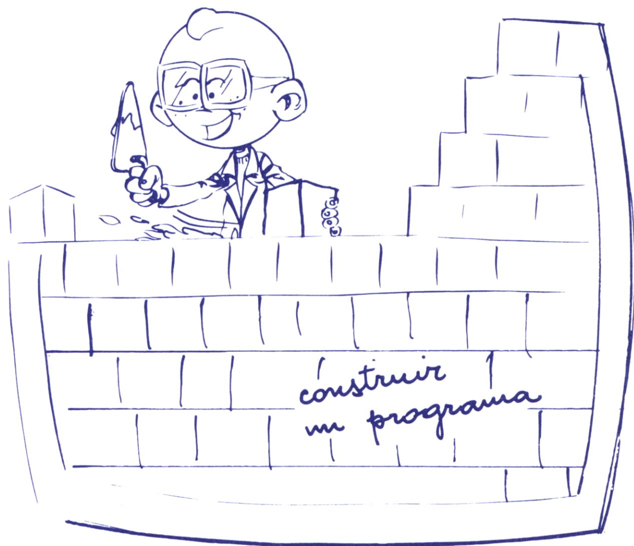
ALGUNOS DETALLES

Mira atentamente esta demostración, te propone simultáneamente un ordinograma, un listado y la ejecución de un programa (lanzamiento de pelota) por tu ordenador. Paso a paso puedes constatar fácilmente las corresponden correspondencias entre esas tres etapas de un programa. Las instrucciones de éste están guardadas unas detrás de las otras por orden creciente de dirección.

Se presentan "de antemano" unas cuantas instrucciones para completar la demostración. Su sintaxis puede modificarse, no la tengas en cuenta.

Lo veremos más tarde.

Hemos descrito detalladamente todas las fases, sin omitir las repeticiones, por ejemplo: después de esta demostración, empezaremos el estudio de la redacción de los programas.


construir
un programa

VARIABLES INDEXADAS
2 programas

N.º ☐

VARIABLES INDEXADAS.

DIMENSIONANDO VARIABLES.

PALABRAS CLAVE

ÍNDICES, DIMENSIÓN, CUADRO.

ALGUNOS DETALLES

La manipulación de un gran número de variables requiere tomar ciertas precauciones: la memoria del ordenador no es ilimitada, por lo cual hay que clasificar las variables dentro de unos cuadros bajo la forma de variables llamadas indexadas.

Se pueden dar índices a las variables cuando tienen una homogeneidad de sentido o de función (todos los vecinos de una calle, todas las pagas de una empresa...). Se reconocerán, entonces, las variables ya no por su nombre sino por su número de índice.

Hay que reservar la localización de esas variables en memoria por DIM (dimensión). La sintaxis de las variables de una dimensión es la siguiente.

DIM A$ (XXX) donde XXX es un número indicando el número de "plazas reservadas".

Para las variables de varias dimensiones, DIM A$ (XXX,YYY,ZZZ) reserva las plazas de la misma manera. Ten cuidado de no reservar demasiadas, porque no quedaría espacio para el programa.

EJERCICIOS Y JUEGOS

Ejercítate en reconocer los lugares de las distintas direcciones del cartero. No es tan fácil.

Busca ejemplos de datos que podrías procesar en 1,2,3,4..., 255 dimensiones.

COMPLEMENTOS SOBRE LOS DATOS

N.º ☐

1 programa

OPERACIONES SOBRE LAS VARIABLES ALFANUMÉRICAS.

EJEMPLOS DE UTILIZACIÓN DE FUNCIONES POCO CORRIENTES.

PALABRAS CLAVE

CONCATENACIÓN.

ALGUNOS DETALLES

Algunas instrucciones permiten modificar cadenas de caracteres (sumar y medirlas...). Es interesante conocerlas bien para la elaboración de ciertos programas. Su sintaxis viene detallada en el manual de tu micro-ordenador.

Algunos ejemplos te demuestran su utilidad. Esas funciones frecuentemente olvidadas son muy útiles y evitan bastantes errores (confusión numérica/alfanumérica...).

Hay que notar que este tratamiento de datos se parece al que ejecuta el ordenador en memoria (recuerda que quita, añade, descuenta..., datos elementales).

Podrás repasar este capítulo de información en el momento del análisis de ciertos problemas por los ejemplos que propone.

GRÁFICOS

N.º ☐

4 programas

PRESENTACIÓN DE LAS INSTRUCCIONES GRÁFICAS.

LOS DISTINTOS MODOS GRÁFICOS.

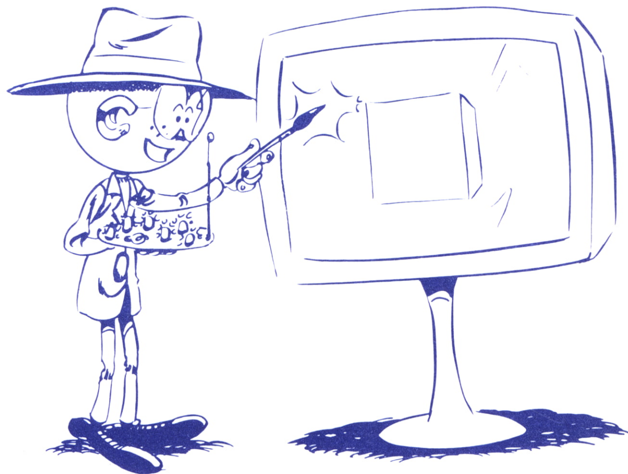
LOS SPRITES.

PALABRAS CLAVE

GRÁFICOS, DIBUJO, MODOS, SPRITE, MEMORIA VDP.

LOS GRAFICOS

Los gráficos es uno de los principales triunfos de los ordenadores MSX, que poseen un circuito especializado capaz de gestionar la pantalla, liberando el microprocesador central.



grafismo

Este circuito se llama "Procesador de vídeo o VDP (Video Display Procesor)". Tiene una memoria propia que contiene todos los gráficos presentes en la pantalla. Vamos a ver en los cuatro programas de este capítulo lo que este procesador puede hacer.

El procesador de vídeo permite utilizar la pantalla de cuatro maneras distintas. Estas cuatro posibilidades permiten conseguir unos gráficos más o menos finos, de más o menos colores en la pantalla.

Esas posibilidades se llaman modos vídeo:

– **El modo TEXTO** permite utilizar una pantalla de 24 líneas con 40 caracteres cada una. De los 16 colores, sólo se pueden utilizar 2. Dispone de un color de fondo y de otro color para los caracteres. Hay 96 caracteres utilizables y representan los alfabetos mayúscula y minúscula. Además se pueden utilizar 160 caracteres suplementarios, alterando el contenido de la memoria vídeo, como lo puedes ver en el primer programa de este capítulo.

– **El modo SEMI-GRAFICO** permite la utilización de una pantalla de 24 líneas con 32 caracteres cada una. Los 16

colores son utilizables simultáneamente en la pantalla, en fondo y forma, lo que representa 256 posibilidades por carácter. Se puede imprimir simultáneamente 256 caracteres diferentes, como en el modo Texto.

El primer programa de este capítulo expone detalladamente, el funcionamiento de estos dos modos, enumerando todas las funciones gráficas del BASIC que son utilizables.

— **El modo ALTA RESOLUCION** permite utilizar una pantalla de 192 líneas con 256 puntos cada una. Se trata de la mejor resolución que se pueda conseguir en un ordenador MSX. Cada punto de la pantalla es direccionable directamente y puede iluminarse con uno de los 16 colores diferentes. Todas las extraordinarias funciones gráficas del MSX se pueden expresar en este modo vídeo.

— **El modo MULTICOLOR** permite utilizar una pantalla con una definición de 64 X 48 puntos. Este modo es poco utilizado, salvo para escribir en una pantalla con letras gigantes, gracias a una herramienta del MSX.

El segundo programa explica el funcionamiento de estos dos modos vídeo.

LOS SPRITES

Abordamos ahora una de las características más interesantes de las instrucciones gráficas. Los SPRITES son unas criaturas que viven dentro del procesador de vídeo. Son 32 y pueden adoptar las formas deseadas y también el color que uno quiere.

El interés de los SPRITES es de representar una forma de varios puntos que uno puede desplazar por medio de una sola instrucción. Cada SPRITE se mueve en un plano diferente al de sus congéneres y existe un orden de aparición en la pantalla.

Todo un conjunto de instrucciones BASIC permite organizar los SPRITES, cambiar sus formas, sus colores y detectar las colisiones entre ellos.

Los SPRITES son utilizables en todos los modos de vídeo salvo en modo multicolor. Pueden tener cuatro tamaños distintos llegando hasta ocupar 32 X 32 puntos. La

finalidad del tercer programa de este capítulo es de enseñarte a manipular fácilmente los SPRITES y a ver hasta qué punto pueden ser útiles en la concepción de los programas, particularmente los de juego.

Finalmente, el cuarto programa de este capítulo te ofrece una herramienta importantísima que te permite definir fácilmente las formas de los SPRITES. Te permite después visualizar los SPRITES, salvaguardar las formas redefinidas y llamarlas ulteriormente.

Estos cuatro programas te permitirán conocer mejor todas las facilidades que te ofrece tu máquina, a nivel de gráficos.

EL SONIDO
1 programa

N.º ☐

PRESENTACIÓN DE LAS INSTRUCCIONES QUE CONTROLAN EL SONIDO.

PROCESADOR DE SONIDO.

PALABRAS CLAVE

PROCESADOR DE SONIDO.

EL MSX ES UN BUEN MÚSICO

De la misma forma que lleva un circuito dedicado a gestión de la pantalla, tu MSX también tiene otro circuito dedicado

al sonido. Este circuito libera al procesador central de todas las tareas relativas al sonido. Permite, por ejemplo, tocar una melodía durante la ejecución de un programa BASIC, sin entorpecer su desarrollo.

Este procesador de sonido sabe hacer muchas cosas a condición de tener paciencia, porque su dominio es un poco difícil de conseguir. Afortunadamente, el BASIC MSC posee varias herramientas que te permitirán componer todas tus melodías. Imagínate que un célebre fabricante de instrumentos consigue transformar tu MSX en un verdadero órgano electrónico sólo con añadir un teclado y un buen programa.

La finalidad de este capítulo es demostrarte que es facilísimo crear melodías sencillas pero que también es posible producir sonidos extremadamente sofisticados, a partir de tu ordenador favorito.

ESTRUCTURA DE PROGRAMACIÓN

N.º ☐

2 programas

ESTRUCTURA DE UN PROGRAMA.

ORGANIZACIÓN DE LAS INSTRUCCIONES.

BUCLE DE CAPTACIÓN.

PALABRAS CLAVE

ANÁLISIS, ORDINOGRAMA, DATOS, CAPTACIÓN.

ALGUNOS DETALLES

PROGRAMA 1

Es la parte crucial de tu iniciación.

Después de haber visto todas las herramientas de las cuales disponemos con el lenguaje BASIC y el tipo de tratamientos informáticos, vamos a empezar la redacción de programas.

La primera parte explica cómo se organiza la ejecución de un programa en función de las órdenes dadas, sin sintaxis. En la segunda parte, se tendrá en cuenta la sintaxis precisa.

Durante la ejecución de ese programa imaginario, descubrirás una nueva función, nunca vista hasta ahora, **GOSUB... RETURN**, es un **GOTO** (instrucción de salto) que permite la ejecución de una acción con una vuelta automática: esta acción se llama subprograma. Así, la orden de impresión es un subprograma en nuestro ejemplo. Esta noción, difícil para los principiantes, se debe conocer, porque es el nervio de los programas complejos que antes de descomponerse en

tareas elementales son analizados en términos de subprogramas menos importantes.

Estos subprogramas se descomponen en tareas elementales. El programa sobre Enrique IV es evolutivo. Lo vamos a complicar poco a poco para hacerlo más difícil, más rico. Empezando por un objetivo sencillo acabaremos con un programa rico que explota casi toda la batería de ordenes BASIC. Debes intentar analizar cada línea de instrucción para entender su posición y su función. No olvides nunca que tu micro-ordenador te repetirá, tantas veces como sean necesarias, una explicación, si tropiezas en un punto concreto.

1.ª ETAPA

Hacemos una pregunta, esperamos la respuesta RE$ que testeamos. Volvemos a hacer la pregunta hasta que la respuesta sea exacta. Si la respuesta es exacta, imprimimos "eso es".

Construye el ordinograma de este problema y observa su ejecución.

2.ª ETAPA

No testeamos una pregunta sino diez. Podríamos testearlas diez veces como antes pero resulta interesante notar la utilización de un bucle FOR... NEXT combinado con un READ... DATA.

O sea el programa:

```
10    FOR I = 1 TO 5
20    READ A
30    PRINT A
40    NEXT I
50    DATA 10, 20, 30, 40, 50
```

Su ejecución da la impresión en la pantalla de 10 20 30 40 50 ¿Qué ocurrió: READ A hace leer el primer dato no utilizado en DATA. Así, a la primera vuelta del bucle A = 10, a la segunda 20...

```
FOR I = 1 TO 5
READ A 1  2  3  4  5  valor de I
NEXT I
DATA 10  20  30  40  50
```

Recuerda esta astucia que permite almacenar datos en memoria y leerlos ahorrando espacio, puesto que los tests y los tratamientos se hacen siempre con las mismas variables (en nuestro ejemplo, en la pantalla Q$, R$).

3.ª ETAPA

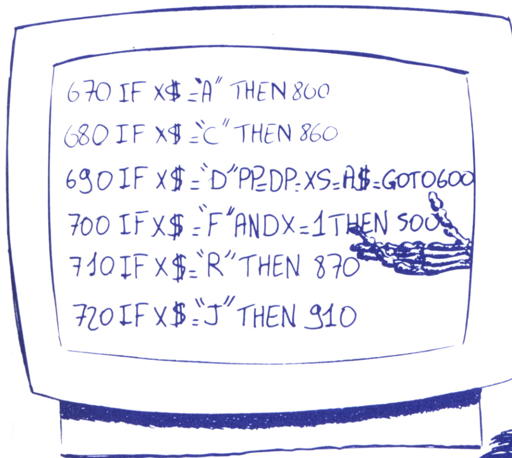
Introducimos varios jugadores. Sus nombres están almacenados en variables indexadas NM$(J) que se definirán por un bucle de captación. Para ayudarte a entender la ejecución, te aconsejamos que apuntes la significación de las variables y que dibujes el ordinograma del problema. Intenta entonces de memoria, sin copiar, volver a escribir el programa y compáralo con el listado real.

Esta simulación te permite dar en el punto clave de un hecho importante de la programación: las posibilidades de evolución de un programa. El juego de preguntas sobre Enrique IV se compone de un "núcleo" que es la adquisición de una respuesta y el test sobre su exactitud. El resto no es más que presentación y facilidad. Hay que intentar siempre resolver primero lo esencial de un problema antes de lanzarse a la escritura irreflexiva de líneas.

EJERCICIOS Y JUEGOS

PROGRAMA 2

Con el fin de participar un poco, el ordenador te ha preparado un ejercicio de elaboración de un repertorio, en el cual debes teclear las instrucciones correspondientes a las órdenes del ordinograma... ¡SUERTE!



ejecución de un programa

ESCRIBIR UN PROGRAMA

N.º ☐

ESCRITURA DE UN PROGRAMA BASIC.

SIMULACIÓN DE EJECUCIÓN.

PALABRAS CLAVE

ORDINOGRAMA, SINTAXIS, LISTADO.

ALGUNOS DETALLES

El ejercicio que se te propone te va a permitir seguir un ordinograma impuesto y teclear tú mismo las órdenes a transmitir. Tus posibles errores serán analizados y comentados, y podrás ensayar cuantas veces te sean necesarias. Si te desanimas, la tecla SELECT te da la solución.

Te aconsejamos que vuelvas a copiar el ordinograma, porque el programa consta de dos partes y corres el riesgo de perder la página... de tu repertorio.

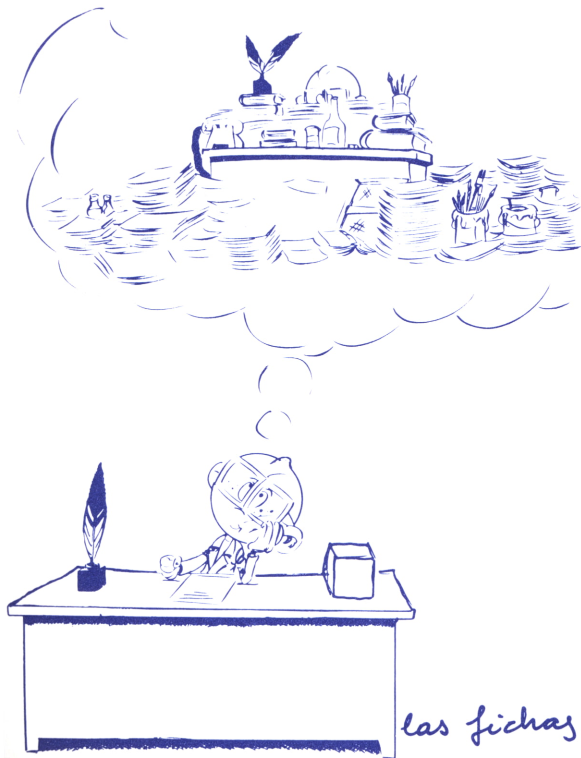
Después de haberlo tecleado asistirás a la ejecución de tu programa. También podrás modificarlo y enriquecerlo. Para ello tendrás que localizar en la "cassette" el punto donde empieza la ejecución del repertorio. Una vez cargado el programa, teclea CTRL+STOP, para la ejecución y teclea LIST, y constatarás que el listado es el mismo que el que tu tecleaste. Entonces podrás tener los valores de los colores, la forma de las letras... e incluso la lógica del programa.

No olvides salvaguardar tu programa modificado en una "cassette" virgen (orden SAVE).

Desde la puesta en funcionamiento de tu mircro-ordenador hemos descubierto las memorias, la manipulación, los datos, la organización, la sintaxis y la construcción de un programa.

Todo parece conseguido... y sin embargo queda lo más difícil. La informática requiere un entrenamiento y una manipulación constante. Nuestro objetivo es sólo una iniciación.

La última parte de nuestra presentación trata de un aspecto fundamental de la informática: los ficheros.


las fichas

FICHERO EXTERNO.

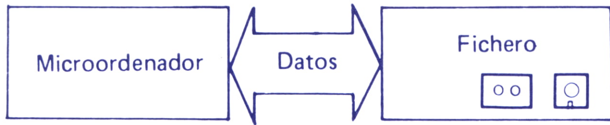
PALABRAS CLAVE

FICHERO.

ALGUNOS DETALLES

Hasta ahora hemos utilizado un "cassette" para leer los programas. También se pueden almacenar datos en "cassette", sin integrarlos en un programa: esto es un fichero externo de datos.

Podemos resumir la situación con el esquema siguiente:



Un programa en memoria puede leer o escribir resultados en una cinta magnética. Este fichero no ocupa más espacio en memoria, pero el acceso a la información es más largo, porque hay que abrir el acceso al fichero, encontrar el dato, obtenerlo y cerrar el fichero: todo lo cual es más lento que buscar en la memoria central.

¿Cuál es el proceso para abrir este fichero?

1/ Autorizar una conexión externa, en escritura (**OUTPUT**) o en lectura (**INPUT**) por **OPEN**.

2/ Utilizar órdenes especiales para escribir o para leer:

INPUT ≠1

PRINT ≠1

El ≠1 caracteriza el fichero y orienta la escritura hacia la cinta o la lectura a partir de la cinta.

3/ Cerrar la conexión con un **CLOSE**.

Esas tres órdenes van siempre juntas. En una cinta magnética significan:

OPEN "CAS":"VINS" FOR OUTPUT AS 1 = abre fichero llamado vinos en escritura (0). Su número es 1.

PRINT 1, "XXX" = escribe XXX en el fichero número 1.
CLOSE 1 = cierra el fichero número 1.

La conexión "cassette"/unidad central es siempre unilateral. No se puede leer y escribir al mismo tiempo en una cinta magnética. Si abres el fichero en lectura o en escritura, tienes que cuidar de enchufar el "cassette", la cinta se pone en marcha y los datos se colocan como signos de programas.

Esos ficheros tienen una ventaja suplementaria: los pueden explotar varios programas. Imaginemos que tú creas un fichero de apellidos y números de télefono, lo puedes explotar en lectura sencilla pero también seleccionar los números, ordenar las personas por número de código telefónico...

El fichero se convierte entonces en una fuente de información. Puedes conservar las "cassettes" para cualquier utilización posterior. Un fichero de DATA puede también conservarse y utilizarse para varios programas (por medio de las fusiones de programas...) pero su manipulación resulta pesada.

En disquetes, un fichero no se organiza tan arcaicamente (secuencial) y permite realizar lecturas/escrituras simultáneas. Las informaciones aquí, están guardadas de otra forma y se puede tener acceso a ellas sin volver a leer todo el fichero (acceso directo).

EJERCICIOS Y JUEGOS

Aquí tienes algunos temas de trabajo para crear ficheros:

— Gestión de tu moto o coche. Construye un fichero del consumo en relación con los kilómetros recorridos. Utilizarás este fichero para editar tus medias y confrontarlas con la media anual.

— Fichero del juego de Enrique IV, visto anteriormente.

BALANCE

Al término de esta inicación, resumamos los principios fundamentales que rigen la creación de todo programa:

ANALIZAR EL PROBLEMA

Hemos visto que el ordenador no sabe gran cosa. Se limita a reconocer los 0 y los 1. Y sin embargo, con los ordenadores hacen el recuento de las elecciones, se calculan las trayectorias de los cohetes... ¿Por qué? Porque cualquier trabajo se puede descomponer en tareas elementales si se analiza correctamente.

INTENTAR SER CLARO Y CONCISO

Las principales órdenes de gestión de las informaciones son sencillas y no debemos complicar los problemas. Bucles, saltos, adquisición... siguen siendo tareas elementales que hay que explotar a fondo.


grafismo

COMENZAR POR LO ESENCIAL

No te pierdas en los programas de presentación de colores... Elije en primer lugar, el "núcleo duro" de tu problema, testeado, y sólo después preocúpate de la presentación.

HAZ UN ORDINOGRAMA

Es un consejo de buen conocedor. Sin plano, uno se pierde en el bosque, incluso si éste parece pequeño.

Y UTILÍZALO

No te limites al ordinograma para el análisis, síguelo para la redacción del programa. Demasiadas veces los principiantes garabatean algunas flechas en un borrador que olvidan por el placer de teclearlo en el teclado. Sin embargo, son dos acciones complementarias. Siempre tiene la posibilidad de titular sus párrafos de listado con la instrucción REM que desaparecerá después en el programa final. Porque REM introduce una frase que no se tiene en cuenta en la ejecución de un programa.

Ejemplo:

10 REM xx calcula Bxx Hacer B = A + 4. 20 B = A + 4

Hacer B = A + 4

20 B = A + 4

REM sólo presenta la acción siguiente y no hay impresión en pantalla. Cuando todo funcione, borrará las líneas REM.

RESPETAR LAS SINTAXIS

Aunque en español existen homónimos cuyos sentidos sabemos distinguir, para un ordenador, una palabra tiene un sólo significado y una sola escritura. Por tanto, no te enfades si no entiende tus órdenes truncadas o tus abreviaciones.

PREVER UNA SALIDA

Una astucia que evita perder el control de los programas: prevé una salida. O bien con END o bien con una elección de tecla "¿Deseas seguir sí-no? ".

SALVAR REGULARMENTE

A medida que escribes tu programa, ten cuidado de salvarlo de forma que en caso de dificultad, te quede una parte importante del trabajo que te resultará fácil volver a leer. Esto evita las largas horas de copia de listados.

Esta serie de consejos debe constituir tu regla de comportamiento permanente durante los primeros programas.

Naturalmente, con el tiempo, llegarás a preferir quizá otros métodos de trabajo, pero en aquel momento ya no necesitarás consejos.

¿Y SI NO FUNCIONA?

Cómodamente instalado delante de la pantalla, ya has tecleado tu programa de previsión de vacaciones, pulsa RUN y... ERROR se imprime en la pantalla. ¿Qué hacer?

La primera comprobación es detectar el tipo de error por el número del mensaje en la pantalla. Una lista de mensajes de tu micro-ordenador te permitirá identificarlo.

Pero, tal vez, ya había cometido una falta grave. **¿HAS SALVADO TU PROGRAMA?** Es posible que hayas perdido todos sus datos y que a partir de entonces el error sea irremediable. Debes volver a escribir todo el listado ¡Qué pena! Antes de la ejecución de todo el programa, te aconsejamos que lo salvaguardes en una "cassette".

El bloqueo de la ejecución de tu programa puede tener tres causas principales:

– Un fallo de lógica.
– Un fallo de sintaxis.
– Un error en la utilización de una instrucción.

En el primer caso no tendrás obligatoriamente errores sino resultados falsos.

Por ejemplo: quieres conocer el precio de un objeto y el ordenador te comunica 20 kilogramos. Vuelve a mirar tu ordinograma y tu listado, una mala conexión, el nombre de una variable utilizada dos veces... la solución no es siempre sencilla y debes proceder paso a paso.

Un fallo de sintaxis es lo más corriente. Generalmente se trata desde la captación del programa donde está escrito el número de la línea errónea. Corrije tu vocabulario.

El fallo en la utilización de una instrucción, mezcla faltas de lógica y de sintaxis. Puede ser un READ sin DATA, un FOR sin NEXT o un READ sin suficientes datos en DATA, una mala medición de variables para DIM...

La metodología es siempre la misma:

COMPROBACION del listado con **LIST** y luego

CORRECCION de los problemas uno por uno. Una vez corregido, no olvides volver a salvar tu programa antes de lanzarlo o de apagar tu ordenador. Sólo te quedaría la antigua versión y habrías trabajado en vano.

CONCLUSIÓN

CHAMPOLLION, con la ayuda de una piedra llena de grafiti llegó a descubrir el secreto de los hieróglifos. Estudiando el lenguaje informático con el cubo, has comprendido que los signos extraños que salen de las máquinas no son tan complejos. De la misma manera que CHAMPOLLION estudió la ciencia de los cómic egipcios y tendió un puente entre nosotros y el pasado, ERNESTO nos ayuda a construir una relación con nuestro futuro... no tan lejano.

A propósito, una última precisión para volver a encontrar a ERNESTO DELTECLADO: antes de desaparecer, miró durante un largo rato la pantalla de nuestro ordenador y dijo: "E. D. llama casa"... Tal vez sea una pista.

INDICE DE LAS PALABRAS CLAVE

ATENCIÓN

En el "cassette" 2 se dice que un ordenador MSX puede tener 16.000 octetos de memoria RAM.

Esta información se refiere a las primeras versiones lanzadas al mercado. En la actualidad la mayoría de equipos MSX disponen de una memoria mayor.

Consulte el manual de su ordenador para conocer las características de su equipo.

En el "cassette" 2 se explica sólo de forma parcial el funcionamiento del comando LINE. En concreto, la opción de rellenar una figura cerrada no se contempla en el ejemplo de prueba.

En ocasiones, el color blanco (código 15) no es utilizable. Se ha hecho así para evitar que pudiera interferirse con los mensajes en blanco que el programa proporciona.



ANAYA
MULTIMEDIA

